

Text Summarization with Pretrained Encoders

Yang Liu and Mirella Lapata

Institute for Language, Cognition and Computation

School of Informatics, University of Edinburgh

`yang.liu2@ed.ac.uk, mlap@inf.ed.ac.uk`

arXiv preprint arXiv:1908.08345 (2019).

背景

- 言語モデルの事前学習は進歩している
 - ELMO, GPT, BERT
- 上記のモデルはさまざまなタスクで使用される
 - 翻訳タスクや文章要約タスクなど
 - 文レベルおよび段落レベルの自然言語理解問題[Devlin 2019]のエンコーダとして採用
- 言語モデルの事前学習が文章要約に与える影響を検証

背景 | 文章要約

抽出型

対象文の中から重要だと思われる文を抽出する手法

- 対象文を要約に含めるべきかどうかの二値分類タスクとして定義
- エンコーダを使用

抽象型

文章の意味を汲み取った上、適切な要約を作成する手法

- 新しい単語やフレーズを含む要約を作成するため言語生成能力が必要
- エンコーダ・デコーダを使用

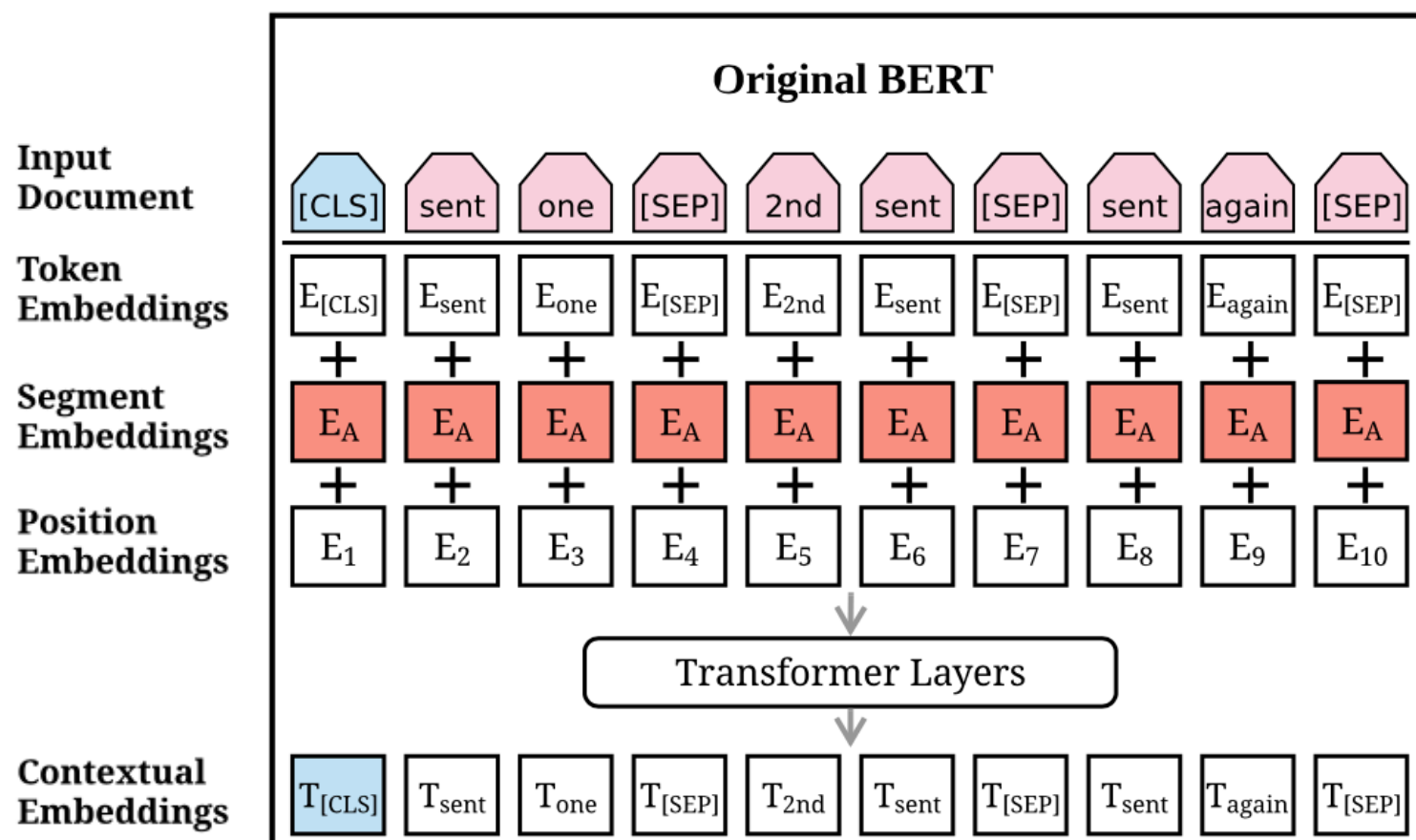
本研究

抽出型および抽象型の両方におけるBERTの可能性を探る

BERT

TransformerのEncoderを使ったモデル

- 入力トークンの一部を[Mask]に置換し元のトークンを当てるタスク
- 2文選んでそれらが連続した文かどうかを当てるタスク
- あらゆるNLPタスクにファインチューニング可能なモデル



- * Transformer : RNNやCNNを用いず, Attentionのみを用いたエンコーダ・デコーダモデル
- * Attention : 文中のある単語の意味を理解する時に文中の単語のどれに注目すれば良いかを表すスコアのこと
- * ファインチューニング : 既存の学習済みモデルの一部と新たに追加したモデルを合わせた全体の微調整を行う. モデル全体のデータを再学習することで汎化性能を向上.

$$\tilde{h}^l = \text{LN}(h^{l-1} + \text{MHAtt}(h^{l-1}))$$

$$h^l = \text{LN}(h^l + \text{FFN}(\tilde{h}^l))$$

$h^0 = x$: 入力ベクトル (3つの総和)

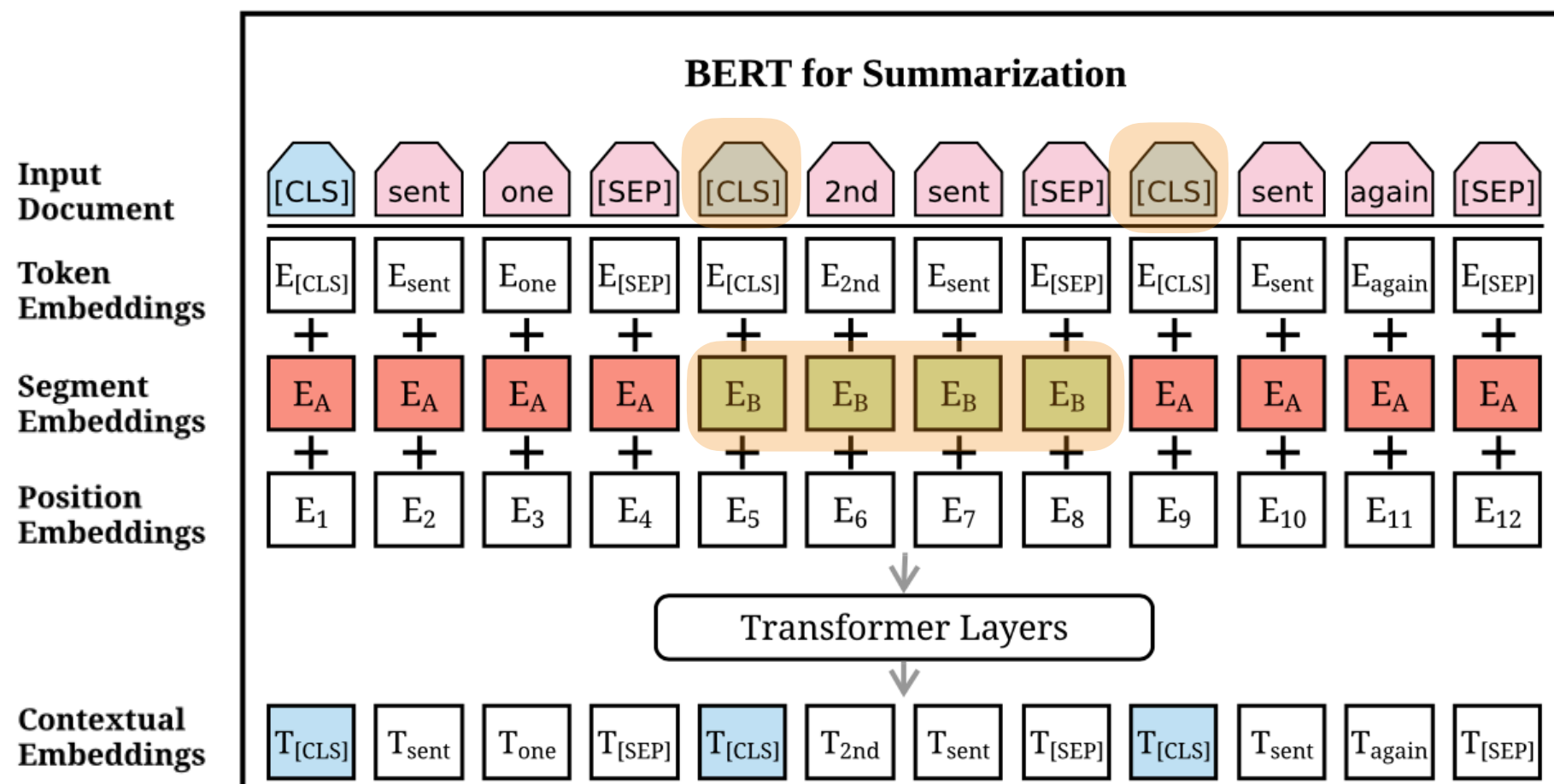
l : Transformer層の深さ

マルチヘッドアテンション演算

Position-wise 順伝播ネットワーク

層の正規化演算

BertSum | BERTをファインチューニング



Token部分とSegment部分を微調整

token : 個々の文を表現するためにCLSトークンを挿入

segment : 文書内の複数の文を区別するため以下のようにする

$$- S[sent_1, sent_2, sent_3] = [E_A, E_B, E_A]$$

BertSum | 抽出型

抽出型

各 $sent_i$ を要約に含めるべきかを示すラベル $y_i \in 0,1$ を付与

$$- h^0 = PosEmb(T)$$

各文の位置を示す正弦波位置埋め込み [Vaswani et al., 2017]]

$$- \hat{y}_i = \sigma(W_o h_i^L + b_o)$$

T : BertSumによって出力された文ベクトル

h_i^L : Transformerの最上層 (L番目) の $sent_i$

シグモイド分類器
(0,1に分類)

実装

Transformer層数 $L = 1, 2, 3$ で実装

$L=2$ が、最も性能が高かった(BertSumExtと呼称)

学習率

$$- lr = 2e^{-3} \cdot \min(step^{0.5}, step \cdot warmup^{-1.5})$$

BertSum | 抽象型

BertSumAbs

エンコーダ：事前学習されたBertSum

デコーダ：ランダムに初期化された6層のTransformer

→ 後者はゼロから学習しなければならないのでミスマッチが生じる

BertSumExtAbs (提案微調整スケジュール適用)

- $\beta_1 = 0.9$, $\beta_2 = 0.999$ の2つのAdam Optimizerを使用

$$> lr_{\epsilon} = \tilde{lr}_{\epsilon} \cdot \min(step^{0.5}, step \cdot warmup_{\epsilon}^{-1.5})$$

$$> lr_D = \tilde{lr}_D \cdot \min(step^{0.5}, step \cdot warmup_D^{-1.5})$$

$$\tilde{lr}_{\epsilon} = 2e^{-3}, \quad warmup_{\epsilon}^{-1.5} = 20000$$

$$\tilde{lr}_D = 0.1 : warmup_D^{-1.5} = 10000$$

定量的評価 | 方法

評価データセット

	文書数	全文の平均長さ	要約の平均長さ	正解集合にのみ 現れるbi-gram
Datasets	# docs (train/val/test)	avg. doc length words sentences	avg. summary length words sentences	% novel bi-grams in gold summary
CNN	90,266/1,220/1,093	760.50 33.98	45.70 3.59	52.90
DailyMail	196,961/12,148/10,397	653.33 29.33	54.65 3.86	52.16
NYT	96,834/4,000/3,452	800.04 35.55	45.54 2.44	54.70
XSum	204,045/11,332/11,334	431.07 19.77	23.26 1.00	83.31

評価指標

- ROUGEを用いて要約の品質を評価
 - > ROUGE-1 : unigram のオーバーラップ
 - > ROUGE-2 : bigram のオーバーラップ
 - > ROUGE-L : 最長共通部分列

定量的評価 | CNN/DailyMail データ

Model	R1	R2	RL
ORACLE → 上限値	52.59	31.24	48.87
LEAD-3 → 最初の3文を選択	40.42	17.62	36.67
Extractive			
SUMMARUNNER (Nallapati et al., 2017)	39.60	16.20	35.30
REFRESH (Narayan et al., 2018b)	40.00	18.20	36.60
LATENT (Zhang et al., 2018)	41.05	18.77	37.54
NEUSUM (Zhou et al., 2018)	41.59	19.01	37.98
SUMO (Liu et al., 2019)	41.00	18.40	37.20
TransformerEXT	40.90	18.02	37.17
Abstractive			
PTGEN (See et al., 2017)	36.44	15.66	33.42
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38
DRM (Paulus et al., 2018)	39.87	15.82	36.90
BOTTOMUP (Gehrmann et al., 2018)	41.22	18.68	38.34
DCA (Celikyilmaz et al., 2018)	41.69	19.47	37.92
TransformerABS	40.21	17.76	37.09
BERT-based			
BERTSUMEXT	43.25	20.24	39.63
BERTSUMEXT w/o interval embeddings	43.20	20.22	39.59
BERTSUMEXT (large)	43.85	20.34	39.90
BERTSUMABS	41.72	19.39	38.76
BERTSUMEXTABS	42.13	19.60	39.18

結果

上限値であるORACLEを除くと
BERTベースが最も精度が高い

提案した学習率を行うと精度が
向上している (一番下の緑部分)

定性的評価

評価方法

生成された要約のみを読んで、関連する質問にどの程度答えられるか
情報量・流暢さ・簡潔さの基準に従って生成された要約2つを比較

Extractive	CNN/DM	NYT
LEAD	42.5 [†]	36.2 [†]
NEUSUM	42.2 [†]	—
SUMO	41.7 [†]	38.1 [†]
Transformer	37.8 [†]	32.5 [†]
BERTSUM	58.9	41.9

Abstractive	CNN/DM		NYT		XSum	
	QA	Rank	QA	Rank	QA	Rank
LEAD	42.5 [†]	—	36.2 [†]	—	9.20 [†]	—
PTGEN	33.3 [†]	-0.24 [†]	30.5 [†]	-0.27 [†]	23.7 [†]	-0.36 [†]
BOTTOMUP	40.6 [†]	-0.16 [†]	—	—	—	—
TCONVS2S	—	—	—	—	52.1	-0.20 [†]
GOLD	—	0.22 [†]	—	0.33 [†]	—	0.38 [†]
BERTSUM	56.1	0.17	41.8	-0.07	57.5	0.19

結果

BERTSUMが最も品質が高いという評価になった

まとめと所感

まとめ

- BERTを要約タスクにファインチューニングした
 - > token部分とsegment部分を調整し、トークン単位ではなく、文単位のベクトル表現を抽出可能に
 - > 抽出型を実装する際のモデルの学習率を調整
- 定性的・定量的評価を行い、どちらも精度が高いことを確認

所感

- transformerやbertの仕組み(数式含め)など、求められる知識が大変多い
- 自分がいま使っているbert as serviceは文を入力するとその文の高次元ベクトルが返ってきてるがこれは文を正確に表せているのか？という疑問

参考資料

- transformer : <https://qiita.com/omiita/items/07e69aef6c156d23c538>
- bert : <https://qiita.com/omiita/items/72998858efc19a368e50>

補足

Transformer

Transformer

RNNやCNNを用いずAttentionのみを用いたエンコーダ-デコーダモデル

- ハイスコア
- 並列化がかなりしやすく訓練時間が圧倒的に削減できる
- 他のタスクにも汎用性が高い

仕組み

Multi-Head Attention（デコーダ部分はカンニングを防ぐためにMask）

- ある1文の単語群のみを使用するSelf-Attention（計算量が少ない）

Position-wise順伝播ネットワーク（各ブロックのAttention層のあと）

- Self-Attention層を通過して形成された位置情報には潜在的にほかの位置情報を含んでいる
→ 依存関係を考慮せずに位置単位で個別に順伝播ネットワークを作り、位置情報があらかず単語情報をユニークなもの（一意なもの）としてとらえることで表現力が増すということから追加

* アテンション：文中のある単語の意味を理解する時に文中の単語のどれに注目すれば良いかを表すスコアのこと